XV6 -Open Source project Assignment-5 Report

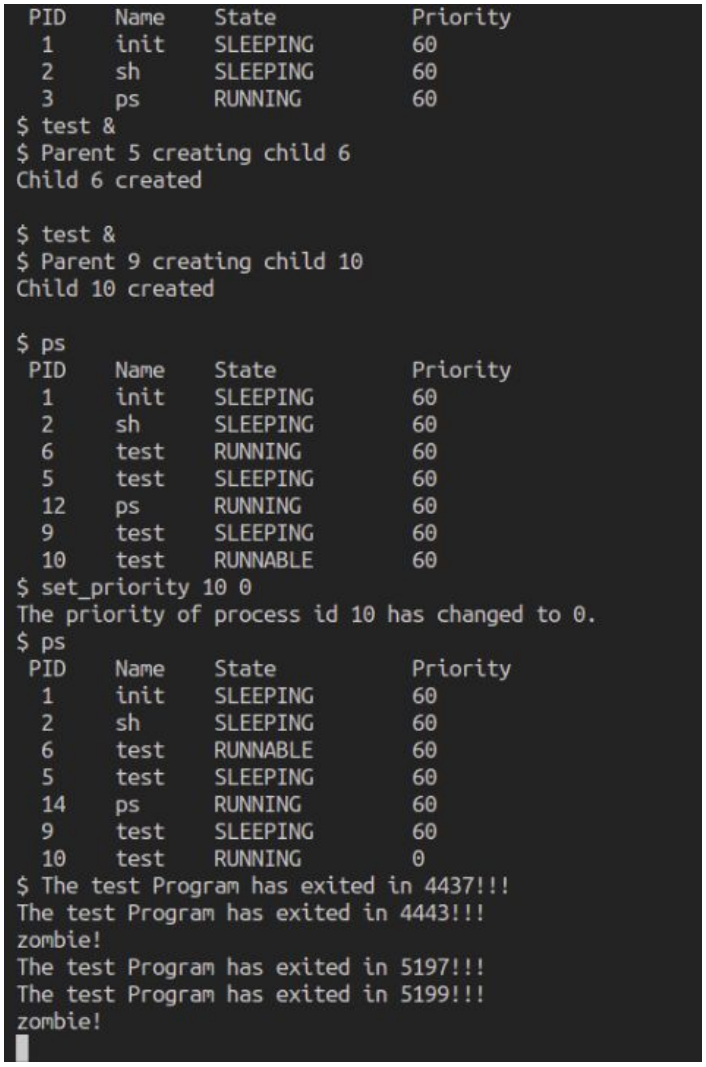
Name - Anurag Sahu (2018121004)

Observations on the Unchanged Xv6 Scheduler :

In the Original Xv6 Scheduler is a simple Round Robin Scheduling, So the average time of completion of the process is large. All the processes have the same priority, they all complete at the towards end. Hence if we initiate two processes one after another they both will get executed simultaneously in round robin and both gets terminated simultaneously.
On the other hand in Implemented Priority based scheduler their are priorities assigned to all the processes, the default priority is set to 60. Smaller the value of priority higher is the priority. This means even if a process has less priority then it will be executed first.
The Test Executes the following code:-

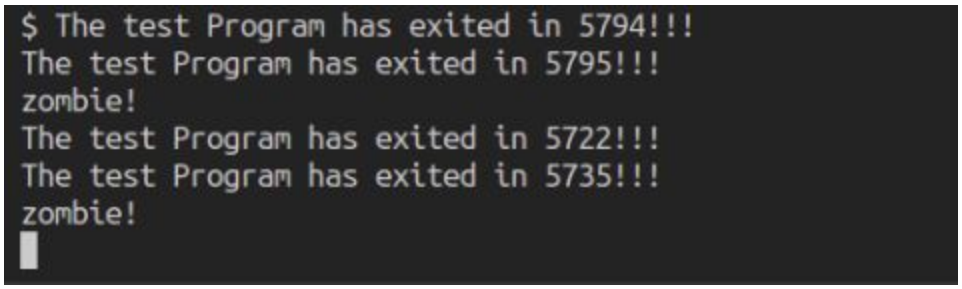
```c
#include "types.h"
#include "stat.h"
#include "user.h"

int
main(int argc, char* argv[])
{
    int k, n = 1, id; // n number of processes that we want to createand
    double x = 0, z, y, d = 1.0;//d is incerment value
    x = 0;
    id = 0;
    int startTime, endTime;
    startTime = uptime();
    for(k = 0; k < n; k++){
        id = fork();
        if(id < 0){
            printf(1,"%d fork failed!\n",getpid());
        } else if(id > 0){
            printf(1,"Parent %d creating child %d\n",getpid(), id);
            wait();
        } else{
            printf(1, "Child %d created\n", getpid() );
            for(z = 0; z < 5000.0; z += d){
                for(y = 0; y < 100000.0; y += d ){
                    x = x + 3 * 4; // useless calculations to comsume CPU time
                }
            }
            break;
        }
    }
    endTime = uptime();
    printf(1,"The test Program has exited in %d!!!\n",endTime - startTime);
exit();
}
```

Here are the screenshot snippets of the test program ran in the priority based scheduler:-



Explanation:- As it can be the clocked units for the different commands is about 600 units. This is because the priority of the second test is changed thus it is executed first, then the second test is executed. Therefore after first test is completed the second test is takes some time to execute then ends.

Here are the screenshot snippets of the test program ran in the Round robin scheduler:-



Explanation:- Here both the processes end at almost same time. This is because here both the process are given same priority and give same time time quantum.